

Natural Buffer Pool

- General Information
 - Setting up a Buffer Pool
 - NATBPSRV Utility to start a Buffer Pool
 - NATBPMON Utility to Monitor or Shut Down a Buffer Pool
 - Invoking NATBPMON
 - NATBPMON Commands
 - Buffer-Pool Trouble Shooting
 - Problem 1 (under OpenVMS)
 - Problem 1 (under UNIX)
 - Problem 2 (under UNIX only)
 - Shutting Down and Restarting the Buffer Pool
 - Dispatch Vector Addresses (OpenVMS/VAX only)
-

General Information

- Introduction
- Object Types
- Coordination under UNIX
- Coordination under OpenVMS
- Multiple Buffer Pools
- Storing Objects in the Buffer Pool
- Restrictions

Introduction

The purpose of the Natural buffer pool is to share Natural objects between several Natural users working on the same computer. It is a pool of storage into which programs compiled with Natural are loaded in preparation for execution. They are moved into and out of the buffer pool as Natural users request Natural objects.

Since Natural generates reentrant Natural object code, it is possible that a single copy of a Natural object can be executed by more than one user at the same time. For this purpose, each object is loaded only once from the system file into the Natural buffer pool, instead of being loaded by every caller of the object.

Object Types

Objects in the buffer pool can be programs, subprograms, maps, global data areas, local data areas, parameter data areas and copycodes. Local data areas, parameter data areas and copycodes are only placed in the buffer pool for compilation purposes.

When a Natural object is loaded into the buffer pool, a control block is allocated for that object. This control block contains information such as the name of the object, what library or application it belongs to, what database ID and Natural system file number the object was retrieved from, and certain statistical information (for example, the number of users who are concurrently executing a program).

Coordination under UNIX

Resource sharing requires that access to the buffer pool is coordinated among all users. Several system resources are necessary to accomplish this. For example, shared memory on the UNIX operating system is used to store the objects and their administrative information. To synchronize access to these objects, a set of semaphores is used. The amount of available shared memory and the number of semaphores is configured statically in the operating system, and as a result, it may be necessary to change system parameters and to recreate the operating system kernel for your installation. Further information about these topics is system-dependent and described in the installation documentation for your UNIX computer.

Coordination under OpenVMS

For OpenVMS, the buffer pool uses a standard locking mechanism which need not be configured by the administrator.

Multiple Buffer Pools

Several instances of the Natural buffer pool can be started on one computer, depending on the individual requirements. It is also possible to run different versions of the buffer pool on one computer without any problems. These buffer pools have nothing in common, except that they run on the same computer.

Storing Objects in the Buffer Pool

When a user executes a program, a call is made to the buffer pool manager. The directory entries are searched to see if the program exists. If it does not exist in the buffer pool, a copy is retrieved from the appropriate library and loaded into the buffer pool.

When a Natural object is being loaded into the buffer pool, a new directory entry is defined to identify this object, and one or more other Natural objects which are currently not being executed may be deleted from the buffer pool in order to make room for the newly loaded object.

For this purpose, the buffer pool maintains a record of which user is currently using which object, and it detects situations in which a user exits Natural without releasing all its objects. It dynamically deletes unused or out-of-date objects to accommodate new objects belonging to other applications.

Restrictions

When using the Natural buffer pool, only minimum restrictions must be considered:

- When a Natural session hangs, do not terminate it by using the OpenVMS command "STOP" or the UNIX command "kill", the terminal command "break" or the "interrupt" key.
If this process is currently performing changes in the buffer pool internal data structures, an interruption may occur at a stage where the update is not fully completed. If the buffer pool internal data structures are inconsistent, this could have negative effects.

Note: This can only happen when the Natural nucleus is executing buffer pool routines.

- All resources must be shared among all users of one Natural buffer pool.
Group membership of a process is used to give access rights for the buffer pool. This means that the shared memory can be changed by all group members, but not by anyone else. The same applies to the semaphores.

Note: All users of the same Natural buffer pool must belong to the same user group on the UNIX operating system. OpenVMS group or system membership depends on how the SYSTEMWIDE parameter is set (0 or 1).

The following applies for OpenVMS only:

- Since on OpenVMS the common resource 'bufferpool' is administrated by the OpenVMS lock manager, each bufferpool on each node has to be addressed by its own cluster-wide unique LCKRESNAME and GSDNAME. To ensure this, the installation procedure inserts for each node where it is executed a new node-specific bufferpool section into the Natural.ini file. Consequently the bufferpool service has to be started on a given node using its specific BPID, therefore the natetc:natbperv_<node>.com was changed. Also each Natural session has to be started using the BPID for the node where the user is currently logged on. To ensure that, the symbol to start Natural contains the string BPID = BP <node> (defined in the LOGIN.COM for Natural). If applications with different priorities are in use, then every application should have its own bufferpool. (System Dead Lock).

Setting up a Buffer Pool

A buffer pool is set up by making various specifications with the NATPARM utility.

To invoke the NATPARM utility, enter "NAT51PARM" at the system prompt. The Natural Parameter Setting menu is displayed on which you first select "Config" and then "Local Configuration File".

Note: If the "Config" option is not displayed on your Natural Parameter Setting menu, this means that you have no permission to modify the configuration files (see the section Administrator Assignments).

In the window that appears, select "Buffer Pool Assignment". A further window is displayed in which you can make the following buffer pool specifications:

General Parameters

- BPNLE - Number of Directory Entries
- Buffer Pool ID (BP or BPID) - Name of the Buffer Pool.
- MAXUSERS - Maximum number of concurrent users.
- MEMSIZE - Size (in MB) of the buffer pool.

Parameters for UNIX Only

- Shared Memory Key
- Semaphore Key

Parameters for OpenVMS Only

- GSDNAME
- LCKRESNAME
- SECTIONFILE
- SFDELETE
- SYSTEMWIDE

Buffer Pool ID (BP or BPID) - Name of the Buffer Pool

Possible values:	<i>name</i> (1 - 8 characters)
Default value:	NATBP

With this parameter, you specify the name (ID) of the buffer pool. BP or BPID can also be specified dynamically.

MAXUSERS - Maximum Number of Concurrent Users

Possible values:	1 - 5000
Default value:	20

This parameter determines the maximum number of users that can have simultaneous access to the buffer pool. This number determines the sizes of some internal tables stored inside the shared memory during startup.

MEMSIZE - Size of the Buffer Pool

Possible values:	1 - 512
Default value:	3

This parameter determines the size (in MB) of the buffer pool. This value is used during startup by the NATBPSRV utility to create a shared memory segment of the specified size.

Shared Memory Key (for UNIX only)

Possible values:	00000000 - FFFFFFFF
Default value:	<i>none</i>

With this parameter, you specify the shared memory key for the IPC facilities that the buffer pool is using. Ask your UNIX administrator to supply the value for this parameter.

Note: The shared memory key must be unique across your entire UNIX system.

Semaphore Key (for UNIX only)

Possible values:	00000000 - FFFFFFFF
Default value:	<i>none</i>

With this parameter, you specify the semaphore key for the IPC facilities that the buffer pool is using. Ask your UNIX administrator to supply the value for this parameter.

Note: The semaphore key must be unique across your entire UNIX system.

SYSTEMWIDE (for OpenVMS only)

Possible values:	0 or 1
Default value:	1

With this parameter, you specify whether the system resources of a buffer pool, such as locks and the global section, are created system-wide or group-wide. Value "1" specifies system-wide, Value "0" specifies group-wide.

Note: It is recommended to set the value of SYSTEMWIDE before starting the buffer pool and to keep the chosen value until the buffer pool has been terminated.

SFDELETE (for OpenVMS only)

Possible values:	0 or 1
Default value:	0

With this parameter, you specify whether the section file is to be deleted when the buffer pool is shut down. Value "0" specifies it is not deleted, value "1" specifies it is deleted.

LCKRESNAME (for OpenVMS only)

Possible values:	any valid filename
Default value:	<i>filename</i>

With this parameter, you specify the name of the lock resource needed by the buffer pool.

GSDNAME (for OpenVMS only)

Possible values:	any valid section name
Default value:	<i>sectionname</i>

With this parameter, you specify the name of the global section to be used by the buffer pool.

SECTIONFILE (for OpenVMS only)

Possible values:	any valid filename
Default value:	NATBPDEV

With this parameter, you specify the file that contains the global section. The default value NATBPDEV is a logical name which is defined when the buffer pool is started by the command procedure "NATETC:NATBPENV_<node>.COM". This command procedure is created during installation.

BPNLE - Number of Directory Entries

Possible values:	10 - 9999
Default value:	10

This parameter determines the number of directory entries. This value is used by NATBPSRV during the start of the buffer pool. If BPNLE has reached the maximum value, this value is automatically increased.

NATBPSRV Utility to Start a Buffer Pool

The NATBPSRV utility is used to start a Natural buffer pool.

Note.

This utility should not be accessible to all Natural users, because it can cause damage to the work of other buffer pool users.

NATBPSRV allocates the resources required by the buffer pool and creates the permanent communication facilities (that is, shared memory and semaphores) used for the buffer pool. The necessary specifications for the resources and facilities are made with the NATPARM utility (as described in the section **Setting up the Buffer Pool** earlier in the section).

This NATBPSRV utility should only be used during system startup, from within the startup file "startup_nat4.com" (for OpenVMS) or "natstart.bsh" (for UNIX).

By default, the buffer pool "NATBP" is started. If another buffer pool is to be started, you specify its name with the following NATBPSRV command line option:

NATBPSRVBP = *buffer-pool-name*

If in the process of starting a buffer pool NATBPSRV discovers that a buffer pool of the same name is already active, it deletes the already active buffer pool. If the deletion fails, NATBPSRV terminates with an appropriate error message.

NATBPMON Utility to Monitor or Shut Down a Buffer Pool

The Natural utility NATBPMON is used to monitor the buffer pool's activity during its operation. NATBPMON can also be used to shut down the buffer pool, when Natural must be stopped on a computer.

Note.

This utility should not be accessible to all Natural users, because it can cause damage to the work of other buffer pool users.

NATBPMON collects information on the current state of your Natural buffer pool.

The buffer pool contains Natural objects (such as maps, programs and subprograms). When an object is invoked, Natural tries to find the object in the buffer pool. If the object is found in the buffer pool, it will be executed without accessing the system file where it is stored. If the object is not found in the buffer pool, the system file containing the object will be accessed, the object copied, and the copy placed in the buffer pool.

All objects in the buffer pool are node-specific and if an object is updated or changed in one node, the object will automatically be deleted from the buffer pool running on this node. But if the same object exists in the buffer pool of another node, the object in the buffer pool of the other node remains unchanged.

Invoking NATBPMON

You invoke the NATBPMON utility by entering a command at the DCL command line.

If the buffer pool "NATBP" is to be used, you invoke NATBPMON as follows:

```
$ NAT51BPMON
```

If another buffer pool is to be used, you invoke NATBPMON as follows:

```
$ NAT51BPMON BP = buffer-pool-name
```

Once you have invoked NATBPMON, the following prompt is displayed:

```
NATBPMON>
```

NATBPMON Commands

NATBPMON provides several commands, that you enter at the NATBPMON> prompt. The individual commands are described below.

CLEAR

CLEAR

This is a synonym of the ZERO command.

CORPSES

CORPSES

The CORPSES command is used to display the list of "corpses".

A "corpse" is an object that has been deleted, but was still being used in the buffer pool when its deletion took place.

Once this object is no longer being used, it will automatically disappear from the list of "corpses".

Note: The column "cusr" (described with the DIR command) indicates if an object is being used.

DELETE

DELETE $\left\{ \begin{array}{c} \textit{pattern} \\ * \end{array} \right\}$

The DELETE command is used to delete an object from the buffer pool.

All objects can be deleted from a buffer pool by using an asterisk (*).

A *pattern* is used to specify a collection of objects, similar to current operating systems which allow the specification of a class of files with wildcards. For details on how to specify a *pattern*, see the DIR command below.

DIR

DIR $\left\{ \begin{array}{c} \textit{pattern} \\ * \end{array} \right\}$
--

The DIR command is used to display a directory listing objects in a buffer pool.

This list of objects contains the following information:

Column	Explanation
indx	A number that the NATBPMON utility automatically assigns to an object when it is loaded into the buffer pool.
cusr	The current number of users that are using an object in the buffer pool.
pusr	The peak number of concurrent activations of an object in the buffer pool.
nusg	The number of times an object has been activated in the buffer pool.
g	Specifies if an object is being loaded into the buffer pool from the system file and will have either of the following values: 0 - The object is not being loaded. 1 - The object is being loaded.
size	Specifies the size (in bytes) of an object in the buffer pool.
key	Specifies the following information about an object: N - The name of the object. L - The library in which the object is located. K - The kind of object (G = generated object module; S = source; D = part of FILEDIR.SAG). T - The object type (which is "blank" in the case of "D" in the K field).

When the DIR command is issued, all objects in the pool will be displayed in a notation similar to the following:

NATBPMON>dir						
indx:	index of the element					
cusr:	current number of concurrent users					
pusr:	peak number of concurrent users					
nusg:	number of usages					
g :	set if object is generating					
indx	cusr	pusr	nusg	g	size	key
1	0	1	4	0	920	(N = "SEL-MAP" L = "DEMO" K = 'G' T = 'M')
2	0	1	2	0	3096	(N = "EMWND" L = "DEMO" K = 'G' T = 'P')
3	0	1	4	0	604	(N = "HDR" L = "DEMO" K = 'G' T = 'P')
4	0	1	7	0	412	(N = "MMUPROG1" L = "RPA" K = 'G' T = 'P')
5	0	1	5	0	372	(N = "MMUPROG2" L = "RPA" K = 'G' T = 'P')
6	0	1	4	0	372	(N = "MMUPROG3" L = "RPA" K = 'G' T = 'P')

To select some objects, it is possible to restrict the values of certain key fields by specifying a matching pattern expression.

To restrict the allowed field values of a given field, the following pattern notation must be used:

name = expression

You can specify multiple patterns by separating them with a comma.

The specified patterns must all match their corresponding fields in order to accept the entire key field.

The expression accepts the specification of the wildcard characters "*" and "?".

The character "*" matches any sequence of characters (also none), and the wildcard character "?" matches exactly one character.

Examples

To select all objects of type "P" in the sample above, the following command would be used:

DIR T = P

To select all programs in the demo library, the following command would be used:

DIR T = P, L = DEMO

To select all objects containing an "M" in their name, the following command would be used:

DIR N = *M*

The special pattern "*" exists, which matches all objects stored in the buffer pool. To select all objects, the following command can be used:

DIR *

DUMP

DUMP

Do not use this command unless you are requested to do so by Software AG Support.

The DUMP command is used for error analysis.

EXIT

EXIT

The EXIT command is used to exit the NATBPMON utility.

HELP

HELP

The HELP command is used to display a list of all available NATBPMON commands.

IPCRM

IPCRM

The IPCRM command is used to free the resources allocated to the buffer pool.

This command should only be issued following a SHUTDOWN command (see below) when there are no active users.

KILL

KILL *n*

The KILL command is used to start the OpenVMS command "stop" from within the NATBPMON utility in a safe way; that is, without the danger of destroying the internal data structure of the buffer pool.

As "*n*" you specify the number of the user to be "killed". This number corresponds to the *index* number as displayed by the WHO command.

PARAM

The PARAM command is used to display the parameter settings for the buffer pool .

Example under Open VMS

```
NATBPMON>param
Active since .....: 25-OCT-2001 11:10:46.14, Version 1.3(435)
Last time cleared .....: 25-OCT-2001 11:10:46.14
Bpid .....: BPALF2
Systemwide .....: 1
Sfdelete .....: 0
Lckresname .....: NATBPLCKV5121_ALF2
Gsdname .....: NATBPV5121_ALF2
Sectionfile .....: NATBPDEV
Memsize .....: 1048576
Maxusers .....: 10
```

QUIT

QUIT

The QUIT command is used to exit the NATBPMON utility; it is a synonym for EXIT.

SHUTDOWN

SHUTDOWN

The SHUTDOWN command is used to shut down the buffer pool.

No new users will be able to use the buffer pool once this command has been issued.

As soon as all users have stopped using the buffer pool, the buffer pool's resources can be deleted with the IPCRM command (see above).

STATUS

STATUS

The STATUS command is used to display statistical information about the buffer pool.

The following statistics are displayed:

```

NATBPMON> status
Active since .....: 25-OCT-2001 11:10:48.35, Version 1.3(435)
Last time cleared .....: 25-OCT-2001 11:10:48.35
Bpid .....: NATBP
Allocated memory (bytes) .....: 1256436 Current users .....:      2
Smallest allocation .....:      20 Peak users .....:      2
Largest allocation .....: 145404 Dead users purged .....:      4
Free memory (bytes) .....: 9229336
Smallest free .....: 15996
Largest free .....: 6355900

Dormant objects .....:      204 Smallest object (bytes) .....:      316
Active objects .....:      0 Largest object (bytes) .....: 43048
Generating objects .....:      0 Total object sizes .....: 1072570
Obsolete objects .....:      0

Attempted locates .....: 126396 Stored objects .....:      0
Attempted fast locates .....: 60116 Loaded objects .....:     3126
Successful fast locates .....: 55528 Activated objects .....: 114323
Percent .....: 92.37 Aborted loads .....:      0

Dormant objects purged .....:      0 Peak parallel activations ...:      2
Object reuse factor .....: 36.57
NATBPMON>

```

The individual statistics are explained in the following tables.

General Information

Active since	Date and time when the buffer pool was started and the version number of the buffer pool.
Last time cleared	Date and time when the buffer pool was most recently cleared.
BPID	Buffer pool ID.

Memory Allocation

Allocated memory (bytes)	Sum total of all allocated memory.
Smallest allocation	Smallest amount of allocated memory.
Largest allocation	Largest amount of allocated memory.
Free memory (bytes)	Sum total of all free memory.
Smallest free	Smallest amount of contiguous free memory.
Largest free	Largest amount of contiguous free memory.

User Statistics

Current users	Number of users currently using the buffer pool.
Peak users	Peak number of users that have been using the buffer pool.
Dead users purged	Number of inactive users, that have been deleted from the buffer pool.

Object Use Statistics

Dormant objects	Number of available, but inactive objects. These objects are in the buffer pool, but are not being used. They are available for later use.
Active objects	Number of active objects. These objects are currently in use by one of the buffer pool users.
Generating objects	Number of objects that are currently being loaded into the buffer pool. These objects will become available as soon as the load operation completes.
Obsolete objects	Number of objects that are to be deleted from the buffer pool, but are still being used. These object can be displayed by using the CORPSES command.

Object Size Statistics

Smallest object (bytes)	Size of smallest object in the buffer pool.
Largest object (bytes)	Size of largest object in the buffer pool.
Total object sizes	Sum total of all objects in the buffer pool.

Locate Statistics

Attempted locates	Number of object locates. This is the number of object activations when the former location of an object was known.
Attempted fast locates	Number of attempted activations with known slot. This is the number of object activations when the former location of an object was known. It is highly probable that an object can be found in the same place in the buffer pool when it is reactivated.
Successful fast locates	Number of successful fast locates.
Percent	Percentage of successful fast locates.

Object Loading Statistics

Stored objects	The number of objects stored in the buffer pool. This is the number of objects that were stored into the buffer pool and which were not loaded from the system file.
Loaded objects	The number of objects loaded from the system file.
Activated objects	The number of objects activated from the bufferpool.
Aborted loads	The number of load operations that were aborted due to memory shortages within the buffer pool, or due to other events.

General Loading Statistics

Dormant Objects purged	The number of unused objects deleted from the buffer pool to make room for newly loaded ones.
Peak parallel activations	The maximum number of parallel activations of one of the objects in the buffer pool.
Object reusage factor	Average number of times an object was reactivated. This number is the ratio of the number of object activations, and the number of objects loaded into the buffer pool.

WHO

WHO

The WHO command is used to display a list of all users who are using the buffer pool.

The following statistics are displayed:

Statistic	Explanation
index	A number that the NATBPMON utility automatically assigns to each buffer pool user.
tid	The user ID, terminal ID and process ID of a process using the buffer pool.

ZERO

ZERO

The ZERO command is used to reset to "0" all counters that are displayed by the STATUS command.

Buffer-Pool Trouble Shooting

This section describes problems that may occur when using the Natural buffer pool and how to solve them.

It is assumed that UNIX users are familiar with the UNIX commands "ipcs" and "adb".

The following are some typical command output examples, with an explanation of what went wrong during their execution. Problem 1 (Natural or the NATBPMON utility cannot be started) and its solution is described for both an OpenVMS and a UNIX environment. Problem 2 and its solution are described only for a UNIX environment.

Problem 1 (under OpenVMS)

Either Natural or the NATBPMON utility cannot be started.

Examples

\$ NATURAL

```
Unable to open Buffer Pool,
Buffer Pool error: "unexpected system call error occurred" (20)
Global shared memory could not be attached.: NATBP5110X_ALFCOM

%SYSTEM-W-NOSUCHSEC, no such (global) section
```

\$ NAT51BPMON

```
Buffer Pool version 1.3(435) of 25-OCT-2001 11:10:52.19
NATBPMON>status
%SYSTEM-W-NOSUCHSEC, no such (global) section
Buffer Pool error: "unexpected system call error occurred" (20)
Global shared memory could not be attached.: NATBP5110X_ALFCOM
NATBPMON>
```

These two examples describe one of the most typical problems you are likely to encounter as a Natural administrator or user. These problems occur when you start Natural or the NATBPMON utility, and the buffer pool is not active.

Solution

Start the buffer pool as described in the example below.

Example:

```
$ @STARTUP_NAT5.COM STARTUP BP
```

Problem 1 (under UNIX)

Either Natural or the NATBPMON utility cannot be started.

Examples

```
hpn::sag 49 => natural bp = sag
```

```

NATURAL V 5.1.1  DEVELOP SAG 2001

NATURAL Startup Error:    16
Unable to open Buffer Pool,
Buffer Pool error:    "unexpected system call error occurred "  (20)
Global shared memory could not be attached.:    shmkey = 11111111

Operating System Error 2 - No such file or directory

```

hpn::sag 50 => natbpmmon bp = sag

```

Buffer Pool version 1.1(198) of 25-OCT-2001 11:10:57.44
NATBPMON>who
Buffer Pool error:    unexpected system call error occurred  (20)
Global shared memory could not be attached.:    shmkey = 11111111
Operating System Error 2 - No such file or directory
NATBPMON>

```

These two examples describe one of the most typical problems you are likely to encounter as a Natural administrator or user. These problems occur when you start Natural or the NATBPMON utility, and the buffer pool is not active.

Solution

Start the buffer pool as described in the section Activating the Natural Buffer Pool.

Use the "ipcs" command to verify the existence of the necessary semaphores and the shared memory:

hpn::sag 51 => ipcs -m -s

```

IPC status from /dev/kmem as of Thu 25-OCT-2001 12:03:24.30
T      ID      KEY      MODE      OWNER      GROUP
Shared Memory:
m      807 0x4e425031 --rw-rw----      sag  natural
Semaphores:
s      85 0x4e425031 --ra-ra----      sag  natural

```

Note: The above output was edited to exclude memory segments and semaphores that do not belong to the Natural buffer pool.

If you cannot find a shared memory segment, or a set of semaphores with the key you assigned them, the buffer pool was not started.

Problem 2 (under UNIX only)

The Natural buffer pool and a Natural utility are not of the same Natural version. If a utility tries to use the buffer pool, the utility and buffer pool versions are checked for equality. If they differ, the access is denied and an error message is output.

Examples

hpn::sag 54 => natural

```

Unable to open buffer pool,
contact your system administrator
bp_error: 25, version mismatch of buffer pool

```

hpn::sag 55 => natbpmn

```
Buffer Pool version 1.1(198) of 25-OCT-2001 12:05:12.06
```

NATBPMON>dir

```
bp_init: res = -1, bp_errno = 25, errno = 0  
buffer pool error message: "version mismatch of buffer pool"
```

Solution

Verify that all utility programs used with the buffer pool are of the same Natural version. To ascertain the versions, use the following commands:

```
hpn::sag 75 = > adb $NATDIR/$NATVERS/bin/natural  
bp_majrel?D  
bp_majrel:  
bp_majrel: 1  
bp_minrel?D  
bp_minrel:  
bp_minrel: 1  
bp_version?D  
bp_version:  
bp_version: 237  
@bp_date?s  
$global$+1718: Thu Oct 25 12:07:58.01 MET 2001  
Ctrl D
```

The sample output above identifies a buffer pool for Version 1.1, which has the sequence version 237. This buffer pool was generated on Thursday, October 25, 2001 at 12:07:58 MET.

The above commands also work with the images NATBPMON and NATBPSRV.

If all programs are of the same Natural version, but contain different buffer pool versions, contact Software AG Support.

Shutting Down and Restarting the Buffer Pool

Usually it should not be necessary to shut down and restart the buffer pool. This may only be necessary if the buffer pool should become unusable due to serious internal errors in the buffer pool, which is extremely unlikely to occur.

If the NATBPMON utility is still able to access the buffer pool, shut down the buffer pool with the SHUTDOWN command.

Once the SHUTDOWN command is executed, new users are denied access to the buffer pool.

After the last user has stopped accessing the buffer pool, buffer pool resources can be deleted by issuing the IPCRM command.

Active buffer pool users can be monitored by issuing the WHO and STATUS commands.

To restart the buffer pool, you call "startup_nat4.com" (under OpenVMS) or "natstart.bsh" (under UNIX) from a sufficiently privileged account.

If the NATBPMON utility is not able to perform a clean shutdown of the buffer pool, the buffer pool must be deleted by using operating system commands. On OpenVMS, it is only possible to rename the sectionfile, because a sectionfile cannot be deleted as long as it is locked by another user. Then restart the buffer pool using "Startup_Nat5.com" and delete the renamed sectionfile after the next reboot.

Note: The rest of this section applies to UNIX environments only.

Use the "ipcs" command to find out the status of the buffer pool's shared memory and semaphores. In the column NATTCH of the output of an "ipcs -m -a" command, you can see the number of processes currently attached to a shared memory segment:

hpn::sag 42 => ipcs -a -m

IPC status from /dev/kmem as of Thu Oct 25 12:15:38.39 2001								
T	ID	KEY	...	OWNER	GROUP	...	NATTCH	SEGSZ
Shared Memory:								
m	707	0x4e425031	...	sag	natural	...	7	153600

It is highly probable that the number of processes attached to shared memory incorporates a Natural nucleus or the NATBPMON utility currently running. Inform the users who run these processes and ask them to terminate them. Or you can terminate them yourself by using the UNIX command "kill", once you have found out their process IDs using the "ps" command.

Once you are sure that no one is using the buffer pool for important work, its resources can be deleted by using the "ipcrm" command:

hpn::sag 43 => ipcrm -M 0x4e425031 -S 0x4e425031

The values specified for the "-M" and "-S" options must be those that were specified inside the parameter module used to start the buffer pool.

Be careful when you delete shared memory and semaphores using the "ipcrm" command. If you accidentally delete the wrong resource, this might have a serious impact on other software products running on your computer.

The result of deletion can be verified by using the "ipcs" command again.

If there are still some memory segments or message queues displayed, they could belong to other software, or they are marked for deletion because some other process is still attached to them.

If the buffer pool cannot be started after removing the shared memory and semaphores, you should consider either rebooting your computer or contacting Software AG Support.

Dispatch Vector Addresses (OpenVMS/VAX only)

Adabas and Natural define dispatch vectors for private system services. These vectors use the change mode codes from -900 to -1024. You should therefore not use these change mode codes for your own purposes.